


# デュアルウェア講習会 II データベースの値を可視化



目標：データベースの値を可視化

# 目次

第 1 章	課題	3
第 2 章	インストール	4
2.1	matplotlib インストール (Windows 版) . . . . .	4
第 3 章	matplotlib の使い方	5
3.1	matplotlib をインポート . . . . .	5
3.2	グラフを描画 . . . . .	5
3.3	データベースから取得した値のグラフ化のサンプルプログラム . . . . .	9
第 4 章	課題	11

温度センサの値を取得して，グラフによる可視化をします（図 1.1）。

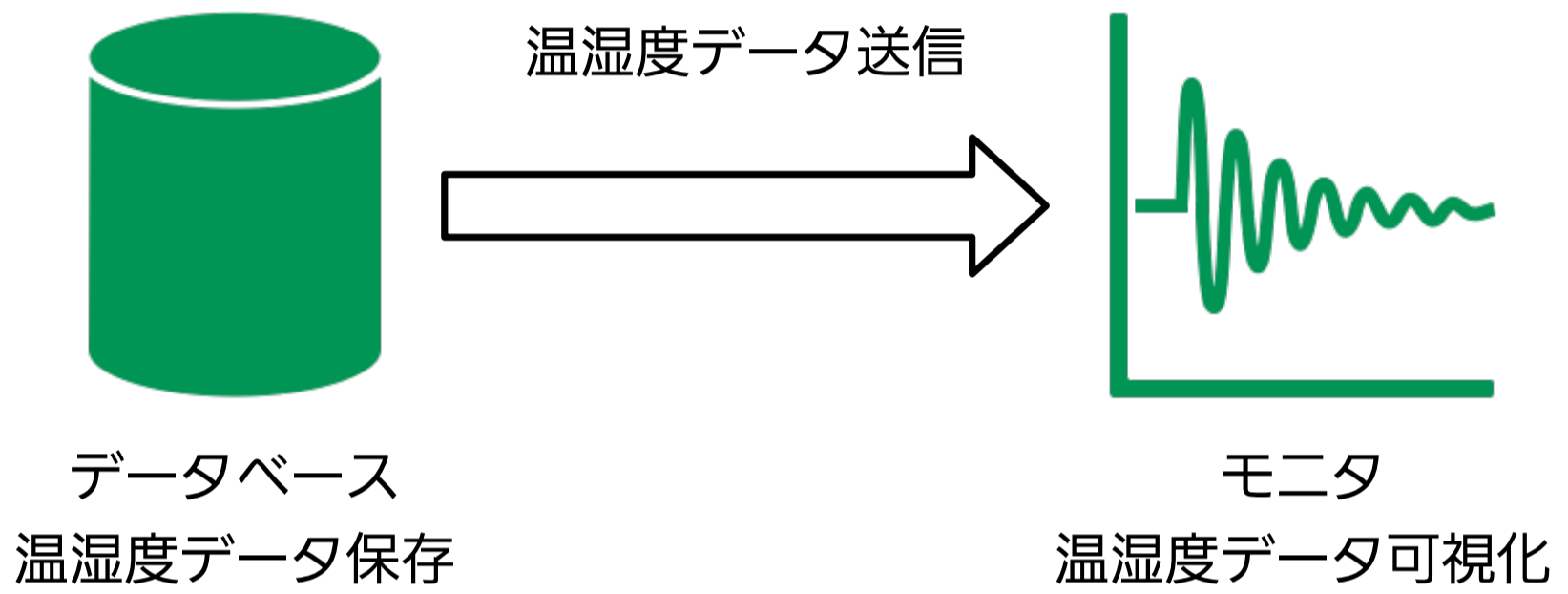


図 1.1: 課題

### 2.1 matplotlib インストール (Windows 版)

#### 2.1.1 numpy インストール

以下のコマンドでインストールをします。

```
1 $ pip install numpy
```

インストールが成功すると、最後に `Successfully installed numpy-1.16.6` <sup>\*1</sup> と表示されます。

インストールされていることを確認します。コマンドプロンプトで `python` と打ち込み、Python の対話モードを立ち上げます。対話モードで `import numpy` を実行し、エラーが表示されないことを確認します。

```
1 $ python
2 >>> import numpy
```

#### 2.1.2 matplotlib インストール

以下のコマンドでインストールをします。

```
1 $ pip install matplotlib
```

インストールが成功すると、最後に `Successfully installed backports.~` と表示されます。

インストールされていることを確認します。コマンドプロンプトで `python` と打ち込み、Python の対話モードを立ち上げます。対話モードで `import matplotlib` を実行し、エラーが表示されないことを確認します。

```
1 $ python
2 >>> import matplotlib
```

---

\*1 1.16.6 はバージョンなので、1.16.6 以外が表示される場合もあります。

matplotlib を使用してグラフを作成します。

### 3.1 matplotlib をインポート

matplotlib を Python で使用するにはパッケージを `import` する必要があります。さらにグラフの描画は matplotlib 内の `pyplot` ライブラリを使用します。したがって、以下のようにモジュールを `import` します。

```
1 import matplotlib.pyplot as pyplot
```

`as` を使用することで、`matplotlib.pyplot` を `pyplot` という名前で使えるようになります。

### 3.2 グラフを描画

#### 3.2.1 グラフを 1 つ描画

最初にグラフを 1 つ描画します。以下のプログラムをコピーして、`04_sample1.py` のファイル名で保存してください。保存後プログラムを実行して、次のような結果になることを確認してください (図 3.1)。

```
1 # -*- coding: utf-8 -*-
2 import matplotlib.pyplot as pyplot
3 x = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
4 y = [1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
5
6 pyplot.plot(x, y, label='y') # 折れ線グラフを作成
7 pyplot.xlabel('x')          # x軸のラベル作成
8 pyplot.ylabel('y')          # y軸のラベル作成
9 pyplot.legend()              # 凡例を作成
10 pyplot.show()                # グラフを描画
```

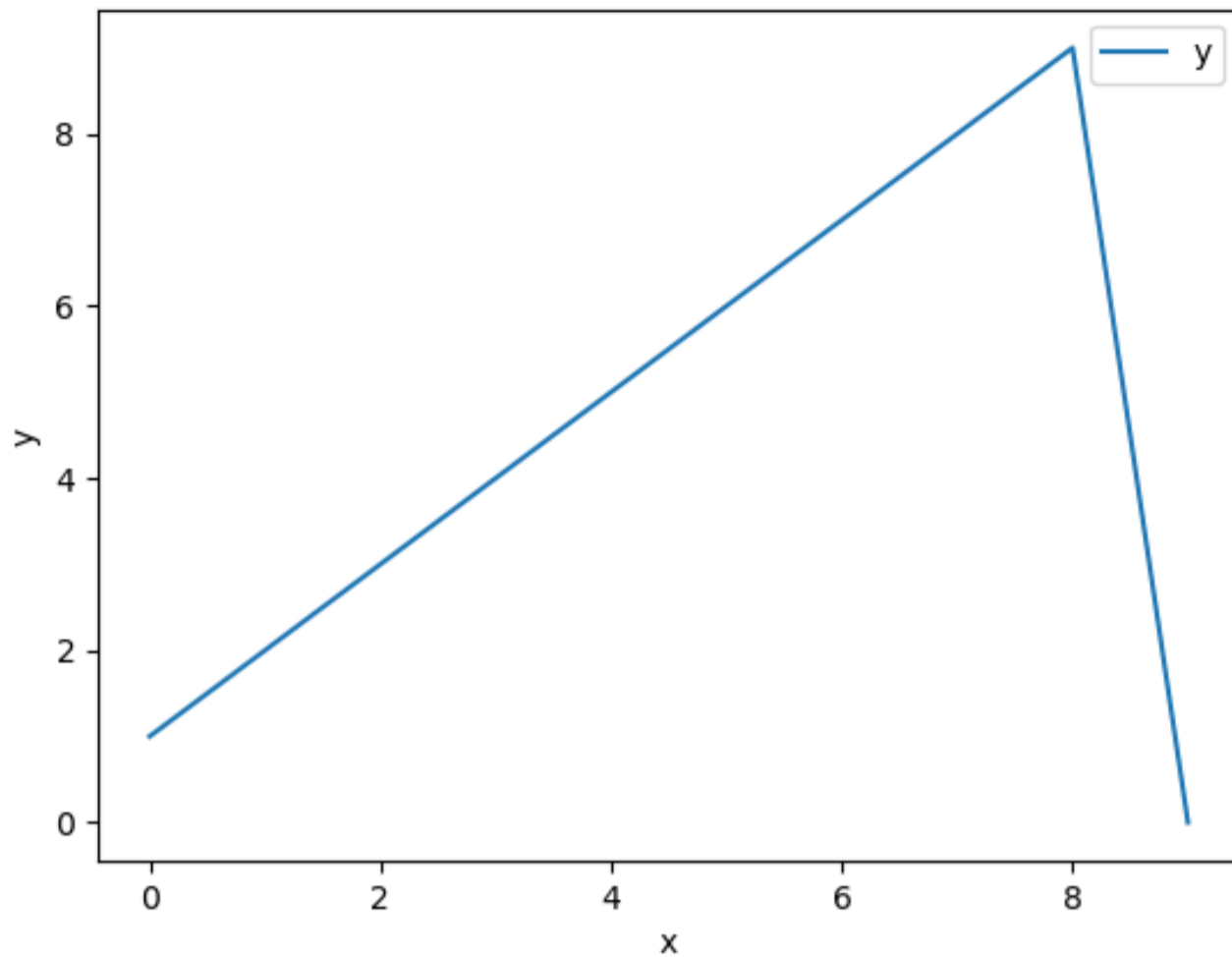


図 3.1: 実行結果

描画に使用したメソッドは以下になります。

plot メソッドは引数のデータで線グラフを作成します (表 3.1)。引数は他にもあり設定できる項目は多くありますが、今回はこの講習会で使用するものを紹介します。

表 3.1: plot メソッド

メソッド名	plot(x, y, label=labelname, color=colorname)	
戻り値	なし	
引数	変数	説明
	x	x 軸の値 (数値リスト)
	y	y 軸の値 (数値リスト)
	labelname	線の名前 (文字列)
	colorname	線の色 (文字列)

以下の xlabel, ylabel メソッドで、グラフの x 軸と y 軸の名前を決めています (表 3.2).

表 3.2: xlabel, ylabel メソッド

メソッド名	xlabel(label)	
戻り値	なし	
引数	引数名	説明
	label	x 軸の軸名 (文字列)
メソッド名	ylabel(label)	
戻り値	なし	
引数	引数名	説明
	label	y 軸の軸名 (文字列)

legend メソッドは、plot メソッドで作成したグラフの名前で凡例を表示します (表 3.3).

表 3.3: legend メソッド

メソッド名	legend()
戻り値	なし

show メソッドはグラフを描画します (表 3.4).

表 3.4: show メソッド

メソッド名	show()
戻り値	なし

plot メソッドはグラフを作成しますが、show メソッドを実行しないと、グラフ描画を行いませんので注意してください。

### 3.2.2 グラフを 2 つ描画

グラフを 2 つ描画したい場合は、plot メソッドを 2 回使用します。

```
1 # -*- coding: utf-8 -*-
2 import matplotlib.pyplot as pyplot
3
4 x = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
5 y1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
6 y2 = [10, 22, 31, 14, 25, 16, 27, 31, 29, 20]
7
8 pyplot.plot(x, y1, label='y1')
9 pyplot.plot(x, y2, label='y2')
10 pyplot.xlabel('x')
11 pyplot.ylabel('y')
```

```
12 pyplot.legend()
13 pyplot.show()
```

プログラムをコピーして、04\_sample2.py というファイル名で実行してください。次のような結果になることを確認してください (図 3.2)。

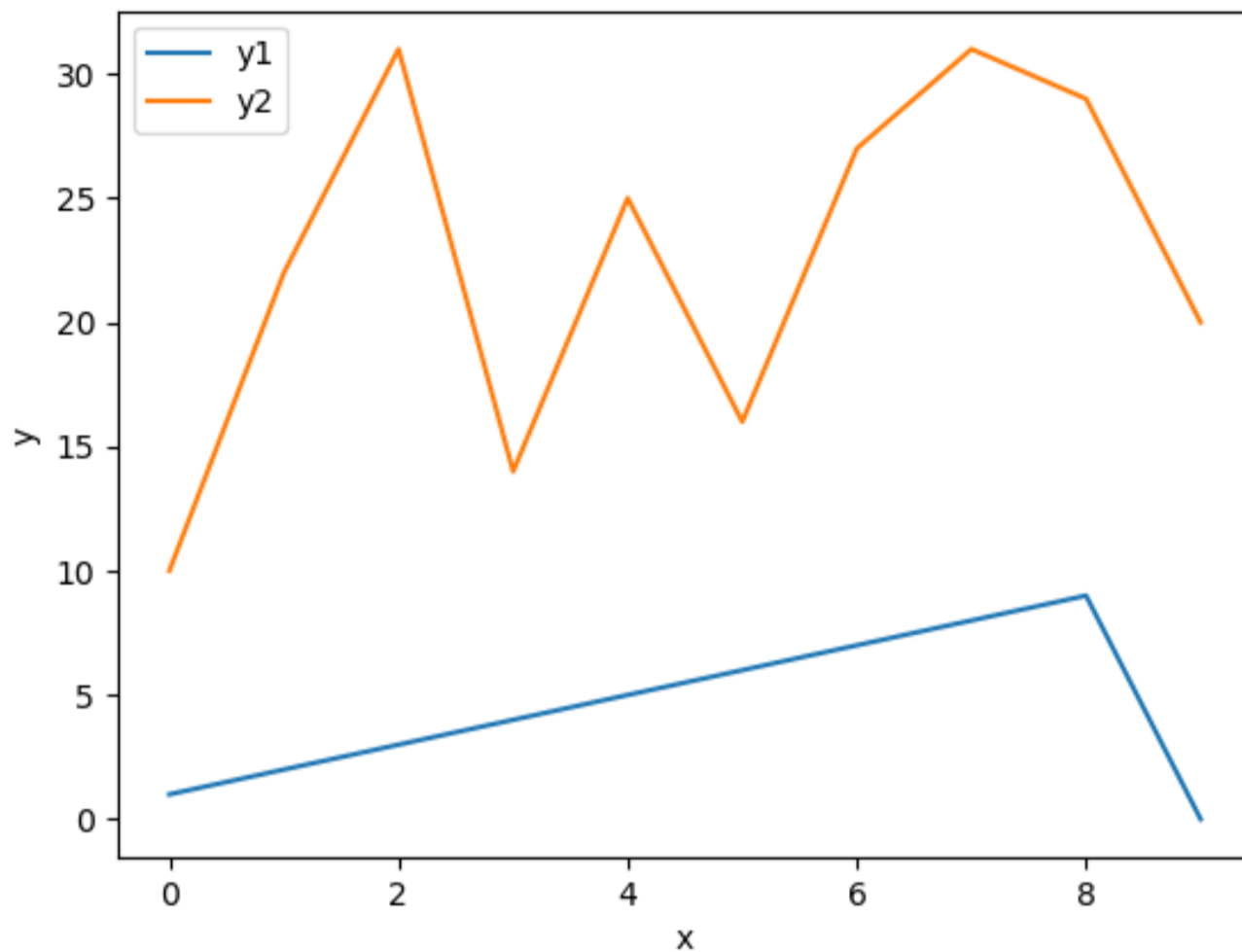


図 3.2: 実行結果

### 3.2.3 グラフを 2 つ並べて描画

グラフを 2 つ並べて描画する場合は以下のようにします。

```
1 # -*- coding: utf-8 -*-
2 import matplotlib.pyplot as pyplot
3
4 x = [0,1,2,3,4,5,6,7,8,9]
5 y1 = [1,2,3,4,5,6,7,8,9,0]
6 y2 = [10,22,31,14,25,16,27,31,29,20]
7
8 pyplot.subplot(2, 1, 1) # 2行1列の1番目
9 pyplot.plot(x, y1,color='blue')
10 pyplot.ylabel('y1')
11
12 pyplot.subplot(2, 1, 2) # 2行1列の2番目
13 pyplot.plot(x, y2,color='red')
14 pyplot.ylabel('y2')
```



## 15 `pyplot.show()`

プログラムをコピーして `04_sample3.py` というファイル名で実行してください。次のような図の結果になることを確認してください (図 3.3)。

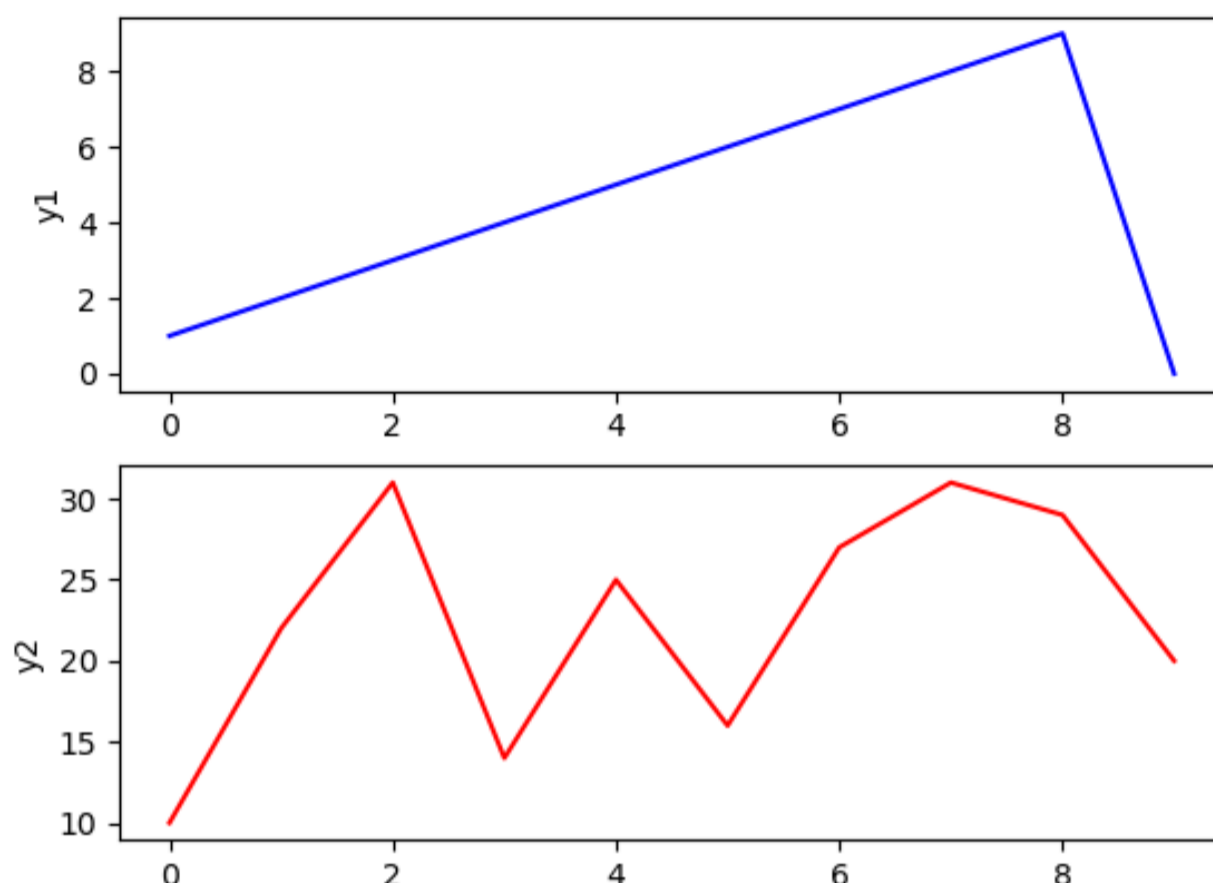


図 3.3: 実行結果

`subplot` メソッドは 1 つの図に収めたいプロットの行数・列数・プロット番号を指定することで、グラフを分割して表示できます (表 3.5)。

表 3.5: `subplot` メソッド

メソッド名	<code>subplot(nrow, ncols, index)</code>	
戻り値	なし	
引数	引数名	説明
	<code>nrow</code>	行数 (整数)
	<code>ncols</code>	列数 (整数)
	<code>index</code>	分割されたウィンドウのどの場所に配置するか番号

引数の `row` と `column` で、ウィンドウを何分割するかを決めます。その分割されたウィンドウを左上から右下に番号付けします。そして、`index` の値でどのウィンドウを使用するかを決めます。たとえば `subplot(2, 1, 1)` とすると、図を 2 行 1 列に分けて 1 番目の場所にグラフを配置するという意味になります。

### 3.3 データベースから取得した値のグラフ化のサンプルプログラム

データベースから温度を取得してグラフ化を行います。

```
1 # -*- coding: utf-8 -*-  
2
```

```

3 # plot
4 import matplotlib.pyplot as pyplot
5
6 # DB data get
7 import psycopg2
8 from datetime import datetime
9
10 # データベース接続パラメータ
11 host = #IPアドレス#
12 dbname = 'pi'
13 user = 'pi'
14 pw = "raspberrry"
15 port = "5432"
16 conect = "host=" + host + " dbname=" + dbname + " user=" + user +
    " password=" + pw + " port=" + port
17 connection = psycopg2.connect(conect)
18 cur = connection.cursor()
19 cur.execute("select * from reidai order by id;")
20 row = cur.fetchall()
21 print(row)
22 x = [] # 時間軸を格納するリストの宣言
23 y = [] # 気温を格納するリストの宣言
24 for i in range(0, len(row)):
25     # date型から文字列型にキャストして格納
26     x.append(row[i][2].strftime('%H:%M:%S'))
27     y.append(float(row[i][3]))
28
29 pyplot.plot(x, y)
30 pyplot.xlabel('time')
31 pyplot.ylabel('temp')
32 pyplot.show()
33 cur.close()
34 connection.close()

```

プログラムをコピーして、04\_sample4.py というファイル名で実行してください。プログラム内の#IP アドレス#を、自分が接続する Raspberry Pi の IP アドレスに変更してください。

以下のプログラムを実行すると、データベースから温度と湿度を取得して、グラフ化による可視化を行います。一部『TODO:』とコメントアウトしていますので、その部分の処理を作成してください。

```
1 # -*- coding: utf-8 -*-
2
3 # plot
4 import matplotlib.pyplot as pyplot
5
6 # DB data get
7 import psycopg2
8 from datetime import datetime
9
10 # データベース接続パラメータ
11 host = #IPアドレス#
12 dbname = 'pi'
13 user = 'pi'
14 pw = "raspberry"
15 port = "5432"
16 conect = "host=" + host + " dbname=" + dbname + " user=" + user +
17         " password=" + pw + " port=" + port
18 connection = psycopg2.connect(conect)
19 cur = connection.cursor()
20 cur.execute(#TODO:SQL文 #)
21 row = cur.fetchall()
22 print(row)
23 x=[] # 時間軸を格納するリスト
24 y1=[] # 気温を格納するリスト
25 y2=[] # 湿度を格納するリスト
26
27 for i in range(0,len(row)):
28     x.append(#TODO:時間の値が入っているリストを指定 #)
```

```
28     y1.append(#TODO: 温度の値が入っているリストを指定 #)
29     y2.append(#TODO: 湿度の値が入っているリストを指定 #)
30
31 pyplot.subplot(#TODO: 上下に2分割して上に表示#)
32 pyplot.plot (#TODO: 温度のx軸#, #TODO: 温度の値#, color='blue', label
    = 'temp')
33 pyplot.ylabel('temp')
34
35 pyplot.subplot(#TODO: 上下に2分割して下に表示#)
36 pyplot.plot(#TODO: 湿度のx軸#, #TODO: 湿度の値#, color='red', label='
    humi')
37 pyplot.ylabel('humi')
38 pyplot.show()
39 cur.close()
40 connection.close()
```

テキストにコピーをして、04\_kadai1.py のファイル名で保存してください。